

Plano de Testes

Desenvolvimento de Sistemas

Professor:

Gracon Huttenberg

Equipe:

Antônio Saraiva
Ana Carolina
Arnold Schwarzenegger
Arthur Pinheiros
Guilherme Hendres

Agosto/ 2022

Histórico de Revisões

Data	Versão	Descrição	Autor
02/08/2022	1.0	Release inicial	Antônio Saraiva
08/08/2022	1.1	Definição dos requisitos a serem testados e designação das estratégias de teste	Ana Carolina, Antônio Saraiva, Arnold Schwarzenegger. , Arthur Pinheiro
11/08/2022	1.2	Organização do documento	Antônio Saraiva
13/08/2022	1.3	Adição de casos de teste	Arnold Schwarzenegger, Guilherme Hendres
15/08/2022	1.3.1	Adição de casos de teste	Arnold Schwarzenegger, Ana Carolina, Antonio Saraiva
16/08/2022	1.3.2	Adição dos testes funcionais de Adição e remoção de consultorias	Arnold Schwarzenegger
17/08/2022	1.3.3	Adição de mais um caso de teste	Antonio Saraiva

1. Introdução	4
1.1 Objetivos	4
1.2 O Avaliauto	4
1.3 Escopo	4
1.4 Identificação de Projeto	4
2. Requisitos a Testar	5
2.1 Teste Unitário - Anexo II	5
2.2 Teste Funcional - Anexo I	5
3. Estratégia de Teste	5
3.1 Tipos de Teste	5
3.1.1 Teste Unitário	6
3.1.2 Teste de Função	6
3.2 Ferramentas	7
4. Recursos	7
4.1 Trabalhadores	7
4.2 Sistema	9
5 Cronograma	9
ANEXO I - Testes funcionais	10
TF1: Login	10
TF2: Excluir Consultoria	11
TF3: Listar Todas as Consultorias	12
TF4: Cadastro de Usuários	13
TF5: Listar Consultorias Finalizadas	15
ANEXO II - Testes Unitários	17
TU1: Solicitar Consultoria	17
TU2: Listar consultorias pendentes	19
TU3: Cadastro de Usuário	20
TU4: Listar Consultorias Aceitas	22
TU5: Listar Consultorias Finalizadas	23
ANEXO III - Gráfico de Cobertura	24

1. Introdução

1.1 Objetivos

O plano de teste do Avaliauto possui objetivo de:

- Verificar as funcionalidades do sistema;
- Ajudar na implementação dos testes;
- Escrever e colocar em prática o que foi visto nas aulas.

1.2 O Avaliauto

O Avaliauto é um sistema online que tem como objetivo auxiliar compradores inexperientes e inseguros na compra de veículos seminovos. Com ele, o comprador pode solicitar a ajuda de um mecânico pelo site para garantir maior segurança na compra de veículos usados. Por outro lado, os mecânicos ganham uma nova oportunidade para lucrar de maneira fácil e rápida.

1.3 Escopo

Esse plano de teste consiste no planejamento, elaboração e execução de testes de caixa branca e preta no sistema Avaliauto a fim de garantir seu bom funcionamento.

1.4 Identificação de Projeto

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Plano de Projeto	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Modelo de Análise	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Modelo de Projeto	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Documento de Arquitetura	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não

Protótipo	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Manual do Usuário	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Lista de Riscos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não

2. Requisitos a Testar

A lista abaixo identifica aqueles itens – use cases, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado.

2.1 Teste Unitário - Anexo II

- TU1: Solicitar Consultoria
- TU2: Listar consultorias pendentes
- TU3: Cadastro de Usuário
- TU4: Listar Consultorias Aceitas
- TU5: Listar Consultorias Finalizadas

2.2 Teste Funcional - Anexo I

- TF1: Login
- TF2: Excluir Consultoria
- TF3: Listar Todas as Consultorias
- TF4: Cadastro de Usuários
- TF5: Listar Consultorias Finalizadas

3. Estratégia de Teste

3.1 Tipos de Teste

Nota: As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

3.1.1 Teste Unitário

Objetivo do Teste	<ul style="list-style-type: none">● Garantir que os models do framework armazenem, recuperem, validem, insiram, ou atualizem os dados.
Técnica	<ul style="list-style-type: none">● Invocar cada método do model do framework que se relacione com alguma funcionalidade a fim de verificar a correta execução dos mesmos.
Critério de Finalização	<ul style="list-style-type: none">● Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.
Considerações Especiais	<ul style="list-style-type: none">● O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para verificar os dados gravados ou lidos do BD.● Processos devem ser invocados manualmente● Bases de dados pequenas ou minimizadas (número de registros limitados) devem ser usados para aumentar a visibilidade de eventos não-aceitáveis.

3.1.2 Teste de Função

Objetivo do Teste	<ul style="list-style-type: none">● Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento, e recuperação.
Técnica	<ul style="list-style-type: none">● Executar cada caso de uso, fluxo de caso de uso, usando dados válidos e inválidos, para verificar o seguinte:<ul style="list-style-type: none">○ Os resultados esperados ocorrem quando dados válidos são usados.

	<ul style="list-style-type: none"> ○ As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados. ○ Cada regra de negócio é aplicada apropriadamente.
Critério de Finalização	<ul style="list-style-type: none"> ● Todos os testes planejados foram executados. ● Todos os defeitos identificados foram tratados.
Considerações Especiais	<ul style="list-style-type: none"> ● Nenhum.

3.2 Ferramentas

As seguintes ferramentas serão empregadas para esse projeto:

Ferramenta	Vendedor	Descrição
TestCase	Django	Classe para testes unitários no Django
Insomnia	Kong Inc.	Cliente para APIs Rest

4. Recursos

Essa seção apresenta os recursos recomendados para o projeto do TIG, suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

4.1 Trabalhadores

Recursos Humanos			
Trabalhador	Recursos mínimos recomendados	Responsabilidade Geral	Responsabilidades Específicas
Gerente de Teste e Gerente do Projeto de Teste	Antônio Saraiva	Fornece supervisionamento gerencial.	Responsabilidades: <ul style="list-style-type: none"> ● Provê direcionamento técnico ● Adquire recursos apropriados ● Fornece relatórios de gerenciamento

Test Designer	Carolina Praxedes	Identifica, prioriza, e implementa os casos de teste.	Responsabilidades: <ul style="list-style-type: none"> • Gera o plano de teste • Cria o modelo de teste • Avalia a efetividade do esforço de teste
Testador	Guilherme Hendres	Executa os testes.	Responsabilidades: <ul style="list-style-type: none"> • Executar os testes • Registrar os resultados • Restabelecer-se dos erros • Documentar solicitações de mudança
Administrador do Sistema de Teste	Antônio Saraiva	Garante que o ambiente e os bens de teste sejam gerenciados e mantidos.	Responsabilidades: <ul style="list-style-type: none"> • administrar o sistema de gerenciamento teste. • instalar e gerenciar o acesso do trabalhador ao sistema de testes.
Gerente do Banco de Dados e Administrador do Banco de Dados	Arthur Pinheiros	Garante que o ambiente e bens de teste de dados (banco de dados) sejam gerenciados e mantidos.	Responsabilidades: <ul style="list-style-type: none"> • administrar os dados de teste (base de dados)
Designer	Arnold Schwarzenegger	Identifica e define as operações, atributos, e associações das classes de teste.	Responsabilidades: <ul style="list-style-type: none"> • identificar e definir as classes de teste • identificar e definir os pacotes de teste
Implementador	Arthur Pinheiro	Implementa e faz os testes unitários das classes e pacotes de teste.	Responsabilidades: <ul style="list-style-type: none"> • cria as classes e pacotes de teste implementados no modelo de teste

4.2 Sistema

A tabela seguinte expõe os recursos do sistema para o projeto de teste.

Recursos do Sistemas	
Servidor de Banco de Dados	<ul style="list-style-type: none">MySQL DataBase Server
Terminais Clientes	<ul style="list-style-type: none">2 PCs (conectados via LAN)1 PC com tela sensível ao toque (conectado a uma LAN e à internet)
Repositório de Testes	<ul style="list-style-type: none">1 PC3 PCs de Desenvolvimento de Teste

5 Cronograma

Milestone	Data de Início	Data de Término
Planejar Teste	08/08/22	08/08/22
Projetar Teste	11/08/22	11/08/22
Implementar Teste	18/08/22	18/08/22
Executar Teste	18/08/22	18/08/22
Avaliar Teste	18/08/22	18/08/22

ANEXO I - Testes funcionais

TF1: Login

Autor: Guilherme Hendres Sousa Melo

Funcionalidade: Login

O que testar: O teste será capaz de testar se o usuário pode se logar caso já esteja cadastrado na plataforma.

Como foi testado: Usando o cliente de APIs Rest Insomnia.

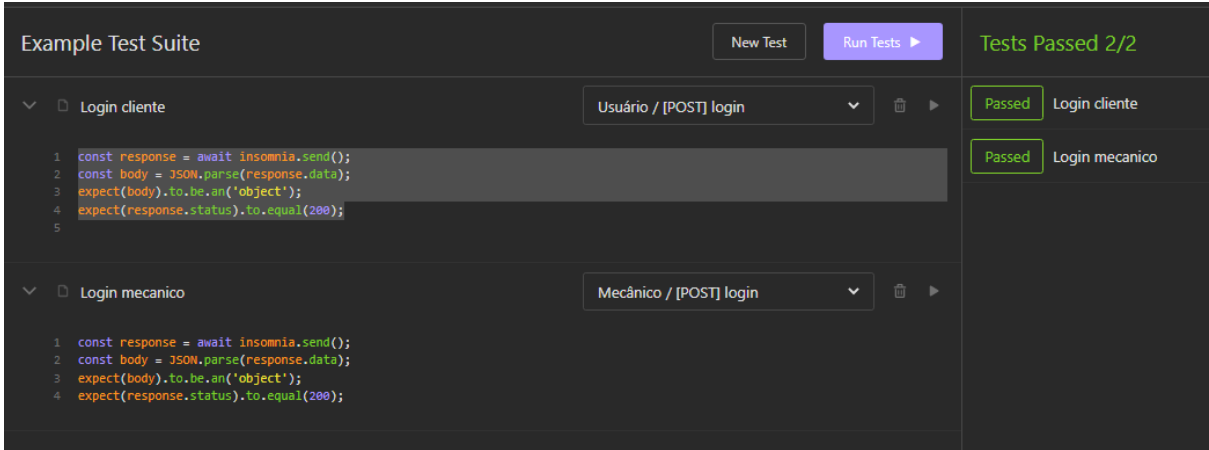
Classes de Equivalência

Variável	Classe	Válida	Inválida
email	existe algum usuário com esse email	1. Sim	2. Não
password	password se refere ao usuário do email	3. Sim	4. Não

Casos de Teste

Código	Classe	Entrada	Saída (status)
CT1	2	email: username@gmail.com senha: 123456	400
CT2	4	email: username arthur.pinheiro@gmail.com senha: 123457	400
CT3	1,3	email: username arthur.pinheiro@gmail.com senha: 123456	200

Resultado:



TF2: Excluir Consultoria

Autor: Arnold S.

Funcionalidade: Excluir consultoria

O que testar: O teste será capaz de testar se o usuário (cliente) consegue excluir uma consultoria pertencente a ele.

Como foi testado: Usando o cliente de APIs Rest Insomnia.

Classes de Equivalência

Variável	Classe	Válida	Inválida
id	existe consultoria com o id	1. Sim	2. Não

Casos de Testes

Código	Classe	Entrada	Saída (status)
CT1	1	id: 2	204
CT2	2	id: 8005	404

Resultado:

Example Test Suite

New Test

Run Tests ▶

Tests Passed 1/1

> □ Login cliente

Usuário / [POST] login

▶

> □ Login mecanico

Mecânico / [POST] login

▶

▼ □ Excluir consultoria

Geral / [DELETE] remover_consultoria

▶

1 const response1 = await insomnia.send();

2 expect(response1.status).toEqual(204);

Passed

Excluir consultoria

TF3: Listar Todas as Consultorias

Autor: Carolina Praxedes

Funcionalidade: Listar Todas as Consultorias

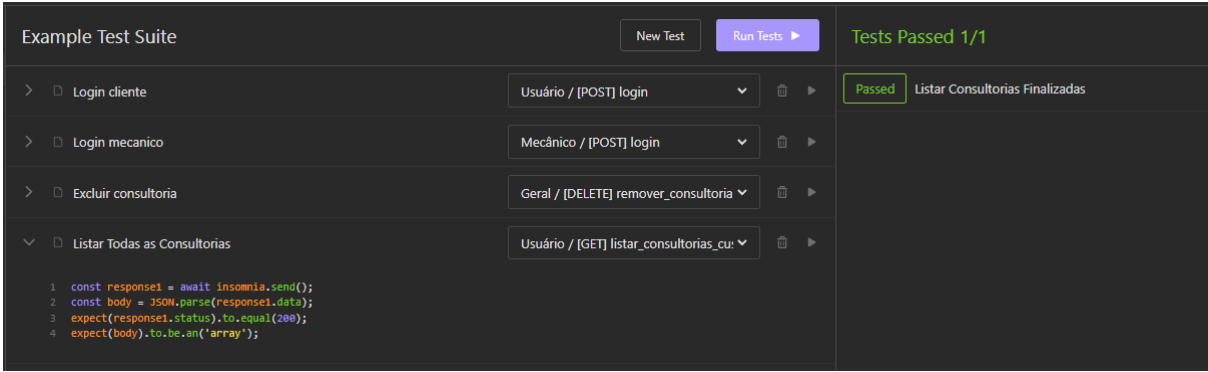
O que testar: O teste será capaz de testar se o usuário consegue visualizar todas as consultorias relacionadas a ele.

Como foi testado: Usando o cliente de APIs Rest Insomnia.

Classes de Equivalência			
Variável	Classe	Válida	Inválida
token	existe usuário cadastrado com o token	1. Sim	2. Não

Casos de Teste			
Código	Classe	Entrada	Saída (status)
CT1	1	token: "Token c4e718707f0323b8b89efd7ce5a27cdd725a89b"	200
CT2	2	token: "Token c4e71870"	401

Resultado:



TF4: Cadastro de Usuários

Autor: Arthur Pinheiro

Funcionalidade: Cadastro de Usuários

O que testar: O teste será capaz de testar se o usuário consegue realizar o cadastro.

Como foi testado: Usando o cliente de APIs Rest Insomnia.

Classes de Equivalência

Variáveis	Condições	Válida	Inválida
password	password com no máximo 255 caracteres	1. password com no máximo 255 caracteres	2. password com mais de 255 caracteres
	password não nula	3. Não nula	4. Nula
email	email com no máximo 255 caracteres	5. email com no máximo 255 caracteres.	6. email com mais de 255 caracteres.
	email não nulo	7. Não nulo	8. Nulo
	email único	9. email único	10. email já pertence a outro usuário
first_name	first_name com no máximo 255 caracteres	11. first_name com no máximo 255 caracteres	12. first_name com mais de 255 caracteres
	first_name não nulo	13. Não nulo	14. Nulo
last_name	last_name com no máximo 255	15. last_name com no máximo 255 caracteres	16. first_name com mais de 255 caracteres

	caracteres		
	last_name não nulo	17. Não nulo	18. Nulo

Casos de Teste

Código	Classe de Equivalência	Entrada	Saída
CT01	2	email: usuario@email.com password: rrrrrrrrrrrr... first_name: usuario last_name: sobrenome	Erro
CT02	4	email: usuario@email.com password: null first_name: usuario last_name: sobrenome	Erro
CT03	6	email: r... .com password: password123 first_name: usuario last_name: sobrenome	Erro
CT04	8	email: null password: password123 first_name: usuario last_name: sobrenome	Erro
CT05	10	email: usuario@email.com (pertence a outro usuário) password: password123 first_name: usuario last_name: sobrenome	Erro
CT06	12	email: usuario@email.com password: password123 first_name: rrrrrrrrrrrr... last_name: sobrenome	Erro
CT07	14	email: usuario@email.com password: password123 first_name: null last_name: sobrenome	Erro
CT08	16	email: usuario@email.com password: password123 first_name: usuario last_name: rrrrrrrrrrrr...	Erro
CT09	18	email: usuario@email.com password: password123	Erro

		first_name: usuario last_name: null	
CT10	1,3,5,7,9,11, 13,15,17	email: usuario@email.com password: password123 first_name: usuario last_name: sobrenome	Cadastro realizado com sucesso

Resultado:

The screenshot shows the Insomnia API client interface. At the top, there's a 'New Test' button and a 'Run Tests' button. Below this, a list of tests is shown under the 'Example Test Suite'. The tests are: 'Login cliente', 'Login mecanico', 'Excluir consultoria', 'Listar Todas as Consultorias', and 'Cadastrar Usuário'. The 'Cadastrar Usuário' test is selected, and its details are shown on the right. The test is a POST request to 'cadastrear_usuario' with a body containing 'Usuário / [POST] login'. The test result is 'Passed'.

TF5: Listar Consultorias Finalizadas

Autor: Antônio Saraiva

Funcionalidade: Listar Consultoria Finalizadas

O que testar: O teste será capaz de testar se o usuário consegue visualizar todas as consultorias relacionadas a ele.

Como foi testado: Usando o cliente de APIs Rest Insomnia.

Classes de Equivalência

Variável	Classe	Válida	Inválida
token	existe usuário cadastrado com o token	3. Sim	4. Não

Casos de Teste

Código	Classe	Entrada	Saída (status)
CT1	1	token: "Token c4e718707f0323b8b89efd7ce5a27cdd725a89b"	200

CT2	2	token: "Token c4e71870"	401
-----	---	----------------------------	-----

Resultado:

Example Test Suite

New TestRun Tests ▶

Tests Passed 1/1

> <input type="checkbox"/> Login cliente	Usuário / [POST] login	Passed	Listar Consultorias Finalizadas
> <input type="checkbox"/> Login mecanico	Mecânico / [POST] login		
> <input type="checkbox"/> Excluir consultoria	Geral / [DELETE] remover_consultoria		
> <input type="checkbox"/> Listar Todas as Consultorias	Usuário / [GET] listar_consultorias_cu:		
> <input type="checkbox"/> Cadastrar Usuário	Geral / [POST] cadastrar_usuario		
▼ <input type="checkbox"/> Listar Consultorias Finalizadas	Usuário / [GET] listar_consultorias_fin		

```
1 const response1 = await insomnia.send();
2 const body = JSON.parse(response1.data);
3 expect(response1.status).toEqual(200);
4 expect(body).toBe.an('array');
```



ANEXO II - Testes Unitários

TU1: Solicitar Consultoria

Autor: Arnold Schwarzenegger Gomes Rodrigues

Funcionalidade: Solicitar Consultoria

O que testar: O teste será capaz de testar se o usuário pode, caso já esteja cadastrado na plataforma, solicitar uma consultoria inserindo as informações necessárias.

Como foi testado: Usando TestCase do Django

Classe de Equivalência

Variáveis	Condições	Válida	Inválida
data_hora	data_hora no formato correto	1. data_hora no formato "2018-10-22 15:00"	2. data_hora difere do formato "2018-10-22 15:00"
	data_hora não nula	3. Não nula	4. Nula
cep	cep com 8 dígitos sem traços	5. cep com tamanho máximo de 8 números (5842135)	6. cep com tamanho > 8 caracteres
	cep não nulo	7. Não nulo	8. Nulo
street	street com tamanho máximo de 255 caracteres	9. street com tamanho máximo de 255 caracteres	10. street com tamanho > 255 caracteres
	street não nula	11. Não nula	12. Nula

Casos de teste

Código	Classe de Equivalência	Entrada	Saída
CT01	2	data_hora: AA/08/2022 14:00 cep: 59100000 número: 555 detalhes: pneu careca ano do veículo: 2010 marca: GM-Chevrolet modelo: Chevrolet Classic street: Prudente de Moraes district: RN district: Natal	Erro

CT02	4	data_hora: null 14:00 cep: 59100000 número: 555 detalhes: pneu careca ano do veículo: 2010 marca: GM-Chevrolet modelo: Chevrolet Classic street: Prudente de Moraes district: RN district: Natal	Erro
CT03	6	data_hora: 12/08/2022 14:00 cep: 591000009 número: 555 detalhes: pneu careca ano do veículo: 2010 marca: GM-Chevrolet modelo: Chevrolet Classic street: Prudente de Moraes district: RN district: Natal	Erro
CT04	8	data_hora: 12/08/2022 14:00 cep: null número: 555 detalhes: pneu careca ano do veículo: 2010 marca: GM-Chevrolet modelo: Chevrolet Classic street: Prudente de Moraes district: RN district: Natal	Erro
CT05	10	data_hora: 12/08/2022 14:00 cep: 59100000 número: 555 detalhes: pneu careca ano do veículo: 2010 marca: GM-Chevrolet modelo: Chevrolet Classic street: Prudente de Moraes... district: RN district: Natal	Erro
CT06	12	data_hora: AA/08/2022 14:00 cep: 59100000 número: 555 detalhes: pneu careca ano do veículo: 2010 marca: GM-Chevrolet modelo: Chevrolet Classic street: null district: RN district: Natal	Erro

CT07	1,3,5,7,9,11	data_hora: 12/08/2022 14:00 cep: 5910000 número: 555 detalhes: pneu careca ano do veículo: 2010 marca: GM-Chevrolet modelo: Chevrolet Classic street: Prudente de Moraes district: RN district: Natal	Cadastro efetuado com sucesso
------	--------------	--	-------------------------------------

Resultado:

```
test_ct01_data_hora_invalid (avaliauto_api.tests.SolicitarConsultoria) ... ok
test_ct02_data_hora_null (avaliauto_api.tests.SolicitarConsultoria) ... ok
test_ct03_cep_max_lenght (avaliauto_api.tests.SolicitarConsultoria) ... ok
test_ct04_cep_null (avaliauto_api.tests.SolicitarConsultoria) ... ok
test_ct05_street_max_lenght (avaliauto_api.tests.SolicitarConsultoria) ... ok
test_ct06_street_null (avaliauto_api.tests.SolicitarConsultoria) ... ok
test_ct07_ok (avaliauto_api.tests.SolicitarConsultoria) ... ok
```

```
-----
Ran 7 tests in 7.049s
```

```
OK
```

TU2: Listar consultorias pendentes

Autor: Ana Carolina Targino Praxedes

Funcionalidade: Listar consultorias pendentes

O que testar: O teste deve verificar se está sendo exibido a lista de consultorias que ainda não receberam uma resposta do mecânico.

Como foi testado: Usando TestCase do Django

Resultado:

```
test_listar_consultorias_pendentes (avaliauto_api.tests.Listar_Consultorias_Pendentes) ... ok
```

```
-----
Ran 1 test in 4.081s
```

```
OK
```

TU3: Cadastro de Usuário

Autor: Arthur Pinheiro

Funcionalidade: Cadastro de Usuário

O que testar: O teste deve verificar se o cadastro de usuário está sendo realizado de forma correta, isto é, verificando se as informações do usuário estão de acordo com o esperado.

Como foi testado: Usando a ferramenta TestCase do Django.

Classe de Equivalência

Variáveis	Condições	Válida	Inválida
password	password com no máximo 255 caracteres	3. password com no máximo 255 caracteres	4. password com mais de 255 caracteres
	password não nula	3. Não nula	4. Nula
email	email com no máximo 255 caracteres	5. email com no máximo 255 caracteres.	6. email com mais de 255 caracteres.
	email não nulo	7. Não nulo	8. Nulo
	email único	9. email único	10. email já pertence a outro usuário
first_name	first_name com no máximo 255 caracteres	11. first_name com no máximo 255 caracteres	12. first_name com mais de 255 caracteres
	first_name não nulo	13. Não nulo	14. Nulo
last_name	last_name com no máximo 255 caracteres	15. last_name com no máximo 255 caracteres	16. first_name com mais de 255 caracteres
	last_name não nulo	17. Não nulo	18. Nulo

Casos de teste

Código	Classe de Equivalência	Entrada	Saída
CT01	2	email: usuario@email.com password: rrrrrrrrrrrr...	Erro

		first_name: usuario last_name: sobrenome	
CT02	4	email: usuario@email.com password: null first_name: usuario last_name: sobrenome	Erro
CT03	6	email: r... .com password: password123 first_name: usuario last_name: sobrenome	Erro
CT04	8	email: null password: password123 first_name: usuario last_name: sobrenome	Erro
CT05	10	email: usuario@email.com (pertence a outro usuário) password: password123 first_name: usuario last_name: sobrenome	Erro
CT06	12	email: usuario@email.com password: password123 first_name: rrrrrrrrrr... last_name: sobrenome	Erro
CT07	14	email: usuario@email.com password: password123 first_name: null last_name: sobrenome	Erro
CT08	16	email: usuario@email.com password: password123 first_name: usuario last_name: rrrrrrrrrr...	Erro
CT09	18	email: usuario@email.com password: password123 first_name: usuario last_name: null	Erro
CT10	1,3,5,7,9,11, 13,15,17	email: usuario@email.com password: password123 first_name: usuario last_name: sobrenome	Cadastro realizado com sucesso

Resultado:

```
test_ct01_password_long (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct02_password_null (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct03_email_long (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct04_email_null (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct05_email_duplicate (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct06_first_name_long (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct07_first_name_null (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct08_last_name_long (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct09_last_name_null (avaliauto_api.tests.CadastroUsuario) ... ok
test_ct10_user_ok (avaliauto_api.tests.CadastroUsuario) ... ok

-----
Ran 10 tests in 8.701s

OK
```

TU4: Listar Consultorias Aceitas

Autor: Guilherme Hendres

Funcionalidade: Listar consultorias pendentes

O que testar: O teste deve verificar se está sendo exibido a lista de consultorias que foram aceitas pelo mecânico mas não foram finalizadas.

Como foi testado: Usando TestCase do Django

Resultado:

```
test_listar_consultorias_aceitas (avaliauto_api.tests.Listar_Consultorias_Aceitas) ... ok

-----
Ran 1 test in 4.098s

OK
```

TU5: Listar Consultorias Finalizadas

Autor: Antônio Saraiva de Gois Junior

Funcionalidade: Listar consultorias Finalizadas

O que testar: O teste deve verificar se está sendo exibido a lista de consultorias que foram finalizadas.

Como foi testado: Usando TestCase do Django.

```
test_listar_consultorias_finalizadas (avaliauto_api.tests.Listar_Consultorias_Finalizadas) ... ok
```

```
-----  
Ran 1 test in 4.082s
```

```
OK
```

ANEXO III - Gráfico de Cobertura

Gráfico de Cobertura

